

МОДЕЛЮВАННЯ ЛОГІЧНИХ ФУНКЦІЙ ДИСКРЕТНОГО ПЕРЕТВОРЕННЯ ІНФОРМАЦІЇ ЗАСОБАМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

В даній статті розглядаються блоки для обчислення ряду операцій та функцій з вхідними та вихідними портами для типів даних signed 32 bits та floating point 32 bits з відповідною таблицею формування тест – кейсів.

Ключові слова: *блок, Simulink, тест – кейс, тип даних, порт, Matlab.*

Вступ. Практичні застосування результатів теоретичних математичних досліджень вимагають отримання відповідей на задачу, що досліджується, в чисельній формі. Це обумовлює проникнення кількісних методів при реалізації досліджень різних сфер людської діяльності за допомогою математичних моделей і перетворення останніх в засоби пізнання. В цьому процесі дослідники прагнуть створити картину, що вимагає необхідності побудови ще більш складних математичних моделей. При дослідженні цих моделей, своєю чергу, часто виникає проблема створення необхідного універсального математичного апарату. Сучасні електронні обчислювальні машини (ЕОМ) дають дослідникам ефективні засоби для цього, а також сприяють ефективному розвитку математичного моделювання складних задач науки та техніки. Реалізація на ЕОМ математичних моделей відбувається на основі цілого ряду нових математичних дисциплін за допомогою методів обчислювальної математики, які постійно вдосконалюється разом з прогресом в області електронно-обчислювальної техніки [1, с. 3].

Математичні моделі дозволяють описати за допомогою математичних залежностей найсуттєвіші характеристики об'єктів чи явищ, тому результати моделювання можуть бути використані при розробці досить складних технічних систем. Зростання складності задач призводить до зростання складності і алгоритмів їх розв'язання на алгоритмічних мовах програмування, не кажучи вже про відлагодження коду, яке займає ще більше часу. Тому в наш час вартим уваги є інший підхід — використання інтегрованих середовищ моделювання.

Серед ряду сучасних програмних систем комп'ютерної математики особливо виділяється математична система MATLAB. Модель створюється з стандартних функціональних графічних блоків, набір яких у пакеті дуже великий і постійно розширюється. Параметри блоків задаються за допомогою зручних діалогових панелей. Результати роботи відображаються або у вигляді графіків, або в цифровій формі для подальшого застосування [2, с. 3].

В Matlab (Matrix Laboratory) основним типом даних є матриця, що робить цю систему особливо зручною для вирішення завдань обчислювальної лінійної алгебри. Система MATLAB є високорівневою мовою і інтерактивним середовищем для розробки алгоритмів, чисельних розрахунків, візуалізації і аналізу даних. Використовуючи Matlab, можна вирішувати обчислювальні завдання набагато швидше, ніж з використанням традиційних мов програмування, наприклад C ++ або Pascal [3, с. 3].

Типове використання Matlab - це:

- 1) математичні обчислення;
- 2) створення алгоритмів;
- 3) моделювання;
- 4) аналіз даних, дослідження та візуалізація;
- 5) наукова і інженерна графіка;
- 6) розробка додатків, включаючи створення графічного інтерфейсу.

Програма Simulink є додатком до пакету Matlab. При моделюванні з використанням Simulink, реалізується принцип візуального програмування,

відповідно до якого, користувач на екрані з бібліотеки стандартних блоків створює модель і здійснює розрахунки. При цьому, на відміну від класичних способів моделювання, користувачеві не потрібно досконально вивчати мову програмування і чисельні методи математики, досить загальних знань, потрібних при роботі на комп'ютері, і, природно, знань тієї предметної області, в якій він працює. Simulink є досить самостійним інструментом Matlab і при роботі з ним зовсім не потрібно знати сам Matlab і інші його додатки. З іншого боку доступ до функцій Matlab і іншим його інструментам залишається відкритим і їх можна використовувати в Simulink. Частина входять до складу пакетів має інструменти, що вбудовуються в Simulink (наприклад, LTI - Viewer додатки Control System Toolbox - пакета для розробки систем управління). Є також додаткові бібліотеки блоків для різних областей застосування (наприклад, Power System Blockset – моделювання електротехнічних пристроїв, Digital Signal Processing Blockset - набір блоків для розробки цифрових пристроїв і т.д). При роботі з Simulink користувач має можливість модернізувати бібліотечні блоки, створювати свої власні, а також складати нові бібліотеки блоків. У ході моделювання є можливість стежити за процесами, що відбуваються в системі. Для цього використовуються спеціальні пристрої спостереження, які, входять до складу бібліотеки Simulink. Результати моделювання можуть бути представлені у вигляді графіків або таблиць. Перевага Simulink полягає також у тому, що він дозволяє поповнювати бібліотеки блоків за допомогою підпрограм написаних як мовою Matlab, так і на мовах C ++, Fortran і Ada [4, с. 10].

Метою даної роботи є створення блоків математичних операцій інформаційно – керуючих систем для візуалізації результатів моделювання на основі програмного пакета Matlab Simulink.

Об'єктом дослідження є моделі при візуалізації процесів моделювання на основі програмного пакета Matlab Simulink.

Предметом дослідження є сукупність теоретичних, практичних основ розробки блоків математичних операцій на основі програмного пакета Matlab Simulink

Методи дослідження. Проведені в роботі дослідження базуються на методах аналітичного моделювання сучасних обчислювальних систем, методи аналізу результатів моделювання.

Для досягнення мети в процесі роботи були сформульовані та розв'язані наступні **задачі**:

- 1) розглянуті теоретичні, практичні аспекти щодо побудови та аналізу математичної моделі;
- 2) побудований функціональний блок;
- 3) побудований математичний блок;
- 4) підібраний набір чисел для вхідних портів (формування тест - кейсів);
- 5) проведена перевірка спрацьовування флажків діагностики;
- 6) проведена перевірка збігу очікуваного результату з даним результатом;
- 7) побудований графік результату програми;
- 8) проведений аналіз отриманих результатів.

Для математичного блоку ми реалізуємо операції додавання, віднімання, множення та ділення. Для функціонального блоку: квадратний корінь числа, модуль, синус, косинус, натуральний логарифм числа, експонента та інверсія числа.

Для початку розглянемо типи даних з якими працюємо. Тип даних single – це числові масиви з числами одинарної точності. Безліч цілих чисел нескінченно, але ми завжди можемо підібрати таке число біт, щоб представити будь-яке ціле число, що виникає при вирішенні конкретної задачі. Безліч дійсних чисел не тільки нескінченно, але ще і безперервно, тому, скільки б ми не взяли біт, ми неминуче зіткнемося з числами, які не мають точного уявлення. Числа з плаваючою комою - один з можливих способів

представлення дійсних чисел, який є компромісом між точністю і діапазоном прийнятих значень.

Число з плаваючою комою складається з набору окремих розрядів, умовно розділених на знак, експоненту порядок і мантису. Порядок і мантиса - цілі числа, які разом зі знаком дають уявлення числа з плаваючою комою.

Математично це записується так:

$$(-1)^s \times M \times B^E$$

де s - знак, B - основа, E - порядок, а M - мантиса.

Основа визначає систему числення розрядів. Математично доведено, що числа з плаваючою комою з базою $B = 2$ (двійкове представлення) найбільш стійкі до помилок округлення, тому на практиці зустрічаються тільки бази 2 і, рідше, 10. Будемо завжди вважати $B = 2$, і формула числа з плаваючою комою матиме вигляд:

$$(-1)^s \times M \times 2^E$$

Мантиса - це ціле число фіксованої довжини, яке представляє старші розряди дійсного числа. Розрізняють нормалізовані та денормалізовані числа.

В нормалізованих числах мантиса завжди позитивна. Якщо ж мантиса не нульова, то число вважається не нульовим, його порядок покладається, причому неявний старший біт мантиси вважається рівним нулю. Такі числа називаються денормалізованими [5].

Для виконання операцій потрібно підбирати такі вхідні дані, щоб спрацьовували усі виключення, тобто `overflow`, `underflow`, `zero`, `nan`, `div_by_zero`.

Overflow (у перекладі з англ. "переповнення") - при встановленому флажку обмеження сигналів цілого типу виконується коректно (див. рисунок 1). Тобто, мається на увазі, щоб результат не виходив за певний діапазон. Якщо результат вийшов за даний діапазон, матимем "+ ∞" або "- ∞".

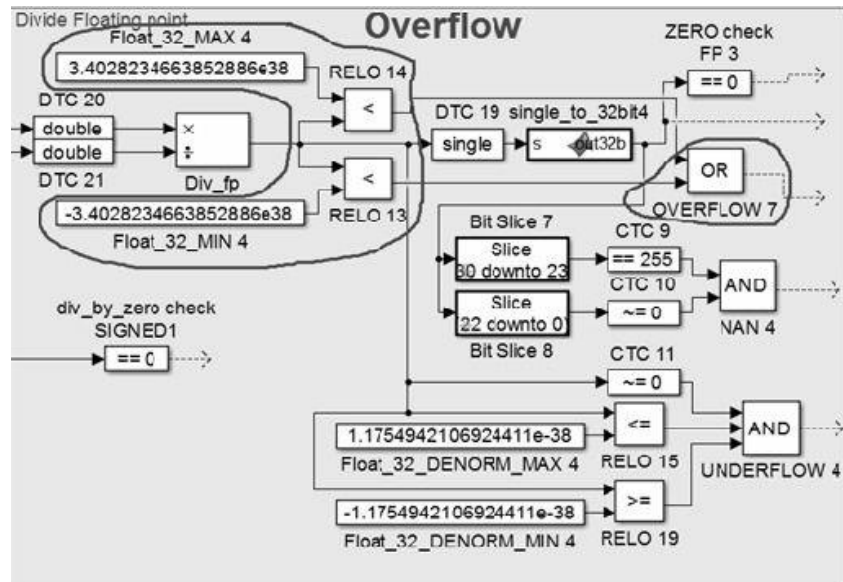


Рис.1. Flag "overflow".

Underflow – результат повинен попасти в діапазон денормалізованих чисел (див. рисунок 2).

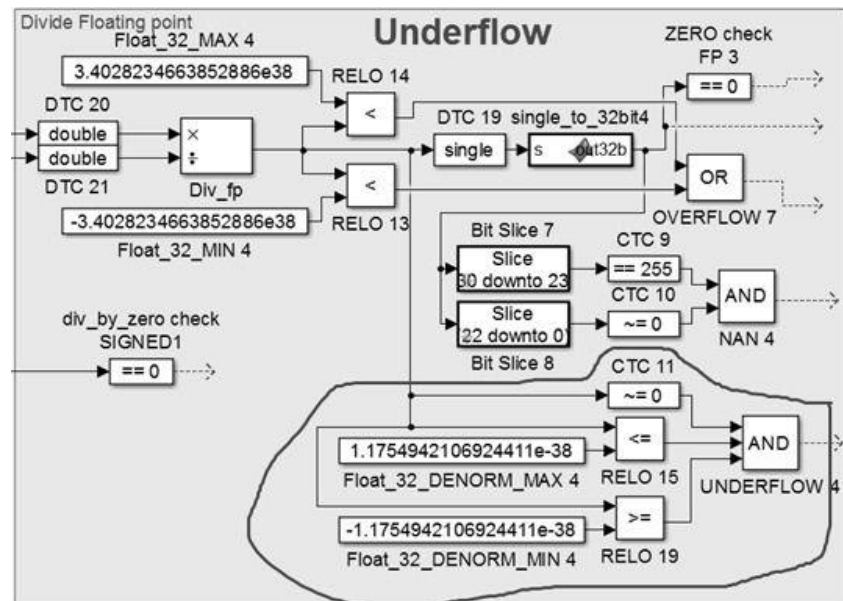


Рис. 2. Flag "underflow".

Zero -перевіряєм чи дорівнює результат нулю (див. рисунок 3).

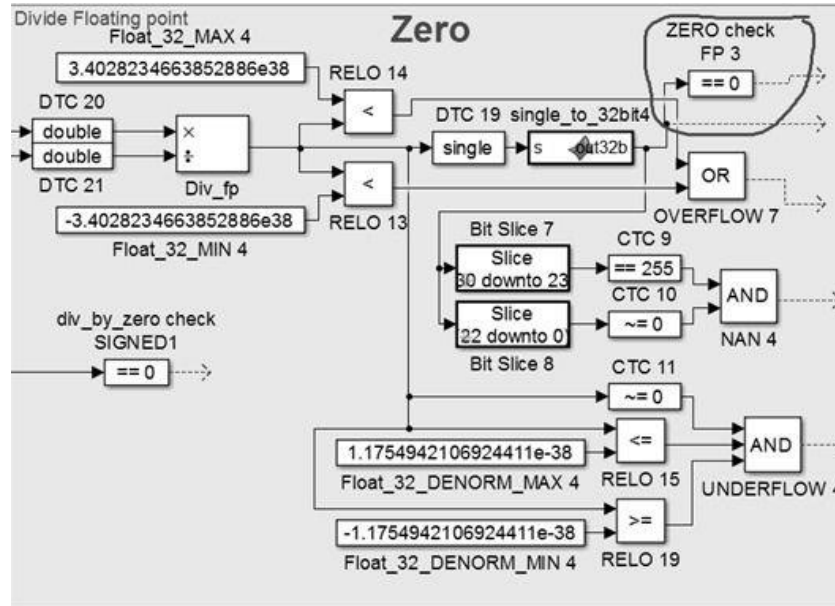


Рис.3. Flag "zero".

NaN – представлення числа, придумане для того, щоб арифметична операція могла завжди повернути яесь значення (див. рисунок 4). У IEEE 754 NaN представлений як число, в якому , а мантиса не нульова. Будь-яка операція з NaN повертає NaN. При бажанні в мантису можна записувати інформацію, яку програма зможе інтерпретувати. Стандартом це не обумовлено і мантиса найчастіше ігнорується.

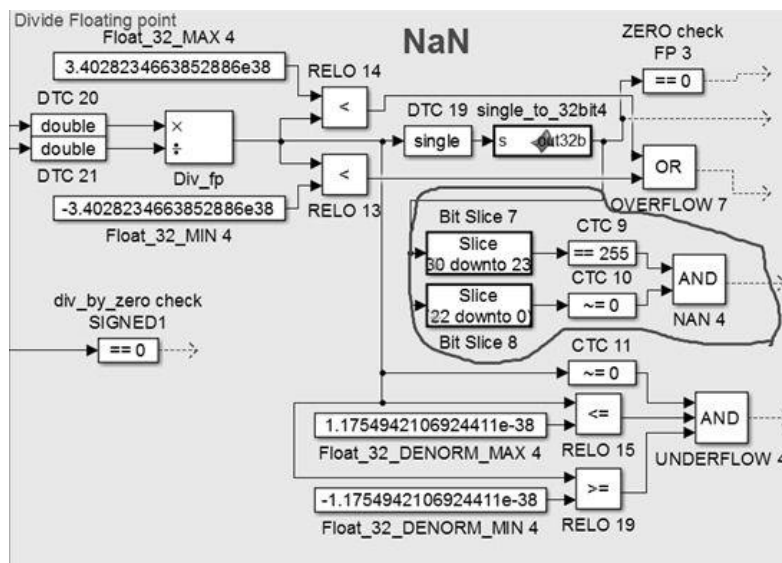


Рис.4. Flag "NaN".

Div_by_zero -перевіряєм ділення на нуль (див. рисунок 5). Результатом буде “+ ”.

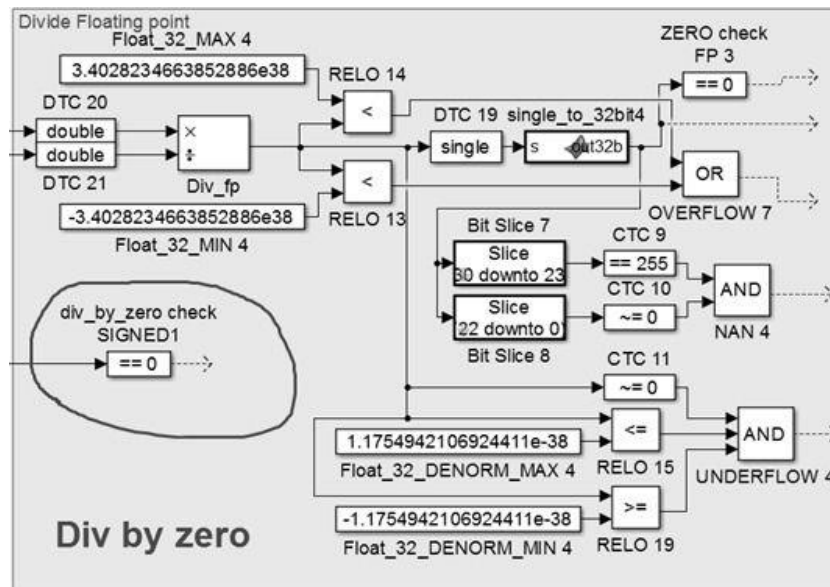


Рис.5. Flag “NaN”.

Для того, щоб передати сигнал від одного блоку до іншого досить просто з'єднати вихід і вхід стрілкою (натискаючи і утримуючи ліву кнопку миші). Тобто сигнал з результатом і флажками діагностики повинні передаватися в блоки "Goto". Для кожного такого сигналу привласнюємо унікальне ім'я, яке відповідає порядковому номеру операції.

Для перевірки роботи математичного блоку на вхід подається 2 операнди: *i_1_oprd* та *i_2_oprd*. Враховуючи, що розглядається два типи даних створюємо Constant *type_of_data*, яка призначена для ідентифікації вхідних даних: *float=1*; *signed=0*. Порт *i_conf* – це ціле додатне число з 5 біт. Параметр конфігурації призначений для того, щоб вибрати потрібну операцію. Тобто, ми повинні підібрати такі два числа, щоб спрацював спочатку один флажок, потім наступні два числа, для спрацювання флажка *overflow* і т.д. Для спрощення роботи сформуємо тест – кейси (сукупність спеціально підібраних вхідних даних для перевірки коректності формування результату). Потрібно враховувати, що для кожної операції ми перевіряємо не всі флажки діагностики. Для типу даних *single* флажок *o_underflow* і *o_NaN* не перевіряються, тому що вони використовуються для чисел з плаваючою

кою. Результат будь-якої операції завжди буде рівен нулю, тому флажок `o_zero` спрацює для кожної операції. Флажок `o_div_by_zero` перевіряється тільки для ділення двох чисел. Переповнення можна утворити для операцій додавання, віднімання та множення. Побудова функціонального блоку є аналогічним до попереднього, відмінність лише в тому, що на вхід подається лише один операнд `i_data` типу `floating point 32 bits`. Далі функція перевірки коректності результату порівнюється з певним числом. Якщо очікуваний результат не співпадає з даним результатом то програма видає помилку, це видно із графіка, коли результат попав в одиничку. Якщо результати збігаються, значить обчислення виконанні вірно, і графік попадає в нуль.

Результат програми видно з графіка блок Score досліджуваних сигналів в процесі моделювання (див. рисунок 6).

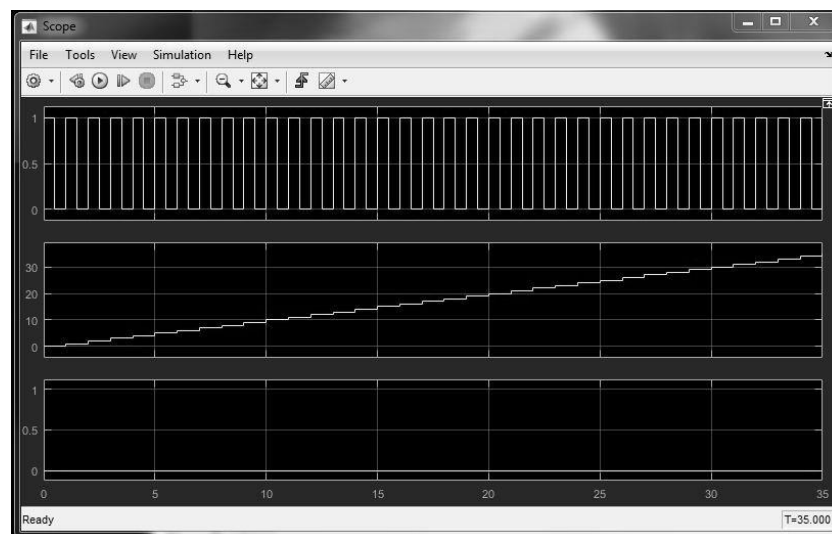


Рис.6. Результат програми.

Перший графік відповідає за тактові часи, тобто в який момент часу робить блок.

Другий графік показує номер прикладу, де 0 – початкове значення і так до останнього прикладу.

Третій графік відображає результат програми. Для того, щоб отримати позитивний результат, графік повинен знаходитись в нулю, якщо попали в одиничку то результат не збігається з очікуваним результатом.

Так як були підібрані правильні вхідні дані і перевірено збіг результату з очікуваним результатом ми отримали позитивний результат.

Висновки Застосування математичної системи MATLAB і пакета імітаційного моделювання Simulink дозволяє швидко, ефективно і надійно вирішувати складні інженерні завдання при проектуванні систем автоматичного управління для реальних динамічних процесів з найрізноманітніших предметних областей. При цьому в залежності від цілей проектування і характеристик об'єкта управління можна використовувати безперервні, нелінійні або дискретні блоки, склад яких постійно розширюється. Передбачена можливість створення власних бібліотек. Глибокі знання з математичного аналізу, тонка інтуїція і відмінні навички застосування сучасних засобів комп'ютерної математики гарантують успіх у справі вирішення складних проблем управління, що виникають в сучасну епоху.

Література

1. *"Комп'ютерні математичні технології" на основі використання математичної системи MATLAB.* Укладач О.Ю. Лісіна. Харків: ХНА-ДУ. 2011. – 85 с.
2. *Сивохин А. В., Мещеряков Б. К. Решение задач оптимального управления с использованием математической системы MATLAB и пакета имитационного моделирования SIMULINK. Лабораторный практикум по основам теории управления.* – Пенза: Изд-во Пенз. гос. ун-та, 2006 – 120с.
3. *Горбаченко В. И.Г67 Вычислительная линейная алгебра с примерами на MATLAB.* — СПб.:БХВ-Петербург, 2011. — 320 с.: ил. — (Учебное пособие) ISBN 978-5-9775-0725-7
4. *Терёхин В.В. Моделирование в системе MATLAB: Учебное пособие /Кемеровский государственный университет.* – Новокузнецк: Кузбассвузиздат, 2004. - 376с.
5. *Что нужно знать про арифметику с плавающей запятой [Электронный ресурс] // – Режим доступа : <https://habrahabr.ru/post/112953/>*

UDC 004.032.26

Bryuhan L.I., student group 601-TI

Odarushchenko O.B., PhD

Poltava National Technical Yuri Kondratyuk University

**DESIGN OF LOGIC FUNCTIONS DISCRETE TRANSFORMATION
OF INFORMATION BY COMPUTER MATHEMATICS**

In this article blocks are examined for the calculation of row of operations and functions with entrance and initial ports for types given of signed 32 bits and floating point 32 bits with the corresponding table of forming test – cases.

Keywords: block, Simulink, test - case, type of data port, Matlab.